1962

Generation and testing of random numbers of an arbitrary distribution

Vaa, Norman A.

Monterey, California: U.S. Naval Postgraduate School

http://hdl.handle.net/10945/12259

GENERATION AND TESTING OF RANDOM NUMBERS OF AN ARBITRARY DISTRIBUTION

NORMAN A. VAA

GENERATION AND TESTING OF RANDOM NUMBERS OF AN ARBITRARY DISTRIBUTION

* * * * *

Norman A. Vaa

NO FORN

GENERATION AND TESTING OF RANDOM NUMBERS OF AN ARBITRARY DISTRIBUTION

by

Norman A. Vaa

Lieutenant Commander, United States Navy

Submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE

United States Naval Postgraduate School
Monterey, California

1962

GENERATION AND TESTING OF RANDOM NUMBERS

OF AN ARBITRARY DISTRIBUTION

by

Norman A. Vaa

This work is accepted as fulfilling

the thesis requirements for the degree of

MASTER OF SCIENCE

from the

United States Naval Postgraduate School

ABSTRACT

Given a method of generating independent random uniform numbers on the unit interval, two methods of generating independent random normal numbers, one method to generate exponential random numbers and one method to generate independent random numbers from the chi-square distribution are presented. All four methods are programmed in FORTRAN on the CDC 1604 high speed digital computer, and tested by a battery of statistical tests. Results of the tests are given in summarized form.

TABLE OF CONTENTS

TABLE OF CONTENTS
(continued)

PREFACE

Contact with some Machine Simulation Models (war games) has indicated that the need for "random numbers" is real and pressing, yet the information available to the war gamers concerning the properties of these numbers is often quite vague and limited.

The average of the results of several runs of a war game is often used to evaluate the many relationships tested by the game. Since random numbers are the device that determines the yes/no answer at the many decision points of the game, the faithfulness with which these numbers exhibit the properties desired by the war gamer directly affect the validity of the results.

In view of the critical role in war gaming played by these numbers it seems useful to provide a tested method of producing the numbers, and to provide also the means of testing any other number-producing methods considered useful.

Random numbers have an important application in statistical sampling, war gaming, and other monte carlo techniques. The applications mentioned in this paper will be primarily in the war game area.

This thesis was written at the United States Naval Postgraduate School, Monterey, California, during the period

January-May, 1962. I am indebted to Professor Jack R. Borsting for his encouragement and capable guidance while acting as faculty advisor, and to Professor Thomas E. Oberbeck for his valuable assistance as second reader. I wish to thank the Naval Postgraduate School Computer Center staff for their patience and efficiency in accepting and processing a multitude of programs compiled during the writing of this thesis.

This thesis was written primarily for the reader with a background in calculus and statistics, but may be useful to anyone with a basic knowledge of the use of computers for war gaming.

INTRODUCTION

The U.S. Navy has procured high speed digital computers and adopted automatic data processing (ADP) to meet the demands of the increasing complexities of modern warfare and weapon systems. In particular, such computers and ADP find application in the Naval Tactical Data System and in Command and Control Centers which have been set up to provide effective direction of military forces. These military activities, with automatic data processing capability, also have a potential capability for war gaming and machine simulation with the computers. Many activities are engaged in the writing, playing and evaluation of war games at this time. It seems likely that war gaming will soon become one of the routine functions of the commands with ADP capability.

To assist in meeting the educational requirements of this situation, formal instruction in war gaming using digital computers is under development at the U.S. Naval Academy and the U. S. Naval Postgraduate School. Fleet Programming Centers have been established for direct support of activities employing automatic data processing and conducting machine simulation war games.

Through the formal instruction in war gaming at the U.S. Naval Postgraduate School it has become apparent that there

are some initial problems in the study of war gaming. One problem lies in having precise terminology and notation to explain concepts that are largely probabilistic in nature. Another lies in the stochastic structure of the game, i.e., the decision points in the game which require drawing a random number to determine the occurrence or non-occurrence of an event. This provides the requirement for a random number generator in the game. A third problem is, given a random number generator, are the characteristics of the numbers produced by this generator such that the results of the game are valid?

The second and third problem areas mentioned, that is, generation of random numbers, and testing of random numbers are the subject of the investigation in this thesis. The investigation included a search of the literature and accumulation of reference material, writing CDC 1604 FORTRAN programs for generation of random normal, exponential and chi-square numbers, writing programs for tests of random numbers, and running the programs for the generators and tests on the CDC 1604 computer.

Sections 1 and 2 of the thesis contain discussions of generation of random numbers and testing of random numbers, respectively. Section 3 contains the results of tests per-

formed on two normal random number generators, an exponential and a chi-square random number generator. Section 4 presents the conclusions reached in this paper. Appendices A, B and C provide justification for the methods of generation of random numbers and Appendix D reproduces the programs in CDC 1604 FORTRAN language for the four generators.

It is suggested that the reader familiarize himself with the explanation of notation and definitions of terms immediately following this introduction. It is also suggested that the U.S. Naval Postgraduate School thesis "Random Number Generation on Digital Computers" by J. M. Barron [9] be consulted for information concerning the generation of uniform random numbers.

DEFINITIONS OF TERMS AND NOTATION

In this paper a random variable, X, and its associated cumulative distribution function F, will be identified as (X,F) or, in some instances, where it is more convenient to refer to the associated density function f, we shall write (X,f). In conformity with standard notation

$$F(x) = Pr(X \leqq x)$$

$$f(x) = \frac{dF(x)}{dx} .$$

Random Numbers: It is considered that there is available some formula or mechanism or device which produces numbers which are regarded as drawn from a population (X,F) or equivalently (X, f). Such a number is frequently identified as a pseudo-random number in the literature; however, here we shall identify it simply as a random number.

We may say that a set of such random numbers is associated with the random variable (X,F) or, more briefly, that they are associated with the distribution F.

Sequence of random numbers: As the formula or the generating device or mechanism employed produces a set of random numbers, these appear serially or sequentially. We shall refer to these numbers as a sequence of random numbers.

In referring to random numbers, as defined above, as-

sociated with specific random variables and their associated cumulative distribution functions, we shall need additional specific definitions and notation.

<u>$N(\mu, \sigma)$</u>: A normal distribution with mean μ and standard deviation σ as parameters.

<u>$U(a,b)$</u>: A uniform distribution over the interval $[a,b]$. In particular, $U(0,1)$ will be used extensively.

<u>Random $N(\mu, \sigma)$ number</u>: A random number associated with the random variable (X,F) where F is $N(\mu, \sigma)$.

<u>Random $U(0,1)$ number</u>: A random number associated with the random variable (X,F) where F is $U(0,1)$.

<u>Normal Deviate</u>: A random $N(0,1)$ number which has been obtained as a transformation of a random $U(0,1)$ number.

<u>Generator</u>: A formula or method for producing a sequence of numbers.

<u>NORMGEN</u>: The computer program to generate normal deviates by the Box-Muller method.

<u>NORMSUM</u>: The computer program to generate normal deviates by the "sums of uniforms" method.

<u>EXPOGEN</u>: The computer program employing Marsaglia's method to generate exponential random numbers.

<u>CHIGEN</u>: The computer program to generate chi-square random numbers from sums of squares of independent normal deviates,

with the number of degrees of freedom as an arbitrary parameter.

SECTION 1

GENERATION OF RANDOM NUMBERS

1.1 BASIC THEORY

For generation of random numbers associated with general random variables (X,F) from random U(0,1) numbers, it is neccessary to establish a relation between the uniform distribution and other distributions. Particularly a relation between the uniform and normal distributions would be useful, because other distributions such as the chi-square, F, and Student's t can be derived from the normal distribution. A general relation between the uniform distribution and other distributions does exist and is called the probability integral transformation. This relation may be stated as follows: (See [8] for a proof)

<u>Theorem</u>: Let X be a continuous random variable with distribution function G. Then the random variable $Y = G(X)$ has a density function given by

$$f(y) = \begin{cases} 1 & 0 \leqq y \leqq 1 \\ 0 & \text{elsewhere;} \end{cases}$$

its distribution function is given by

$$\Pr\left[Y \leqq y\right] = \begin{cases} 0 & y \leqq 0 \\ y & 0 < y < 1 \\ 1 & y \geqq 1 \end{cases}$$

The relation $y = G(x)$ is called the probability integral transformation, and since G is a non-decreasing function it follows that (see page 313 of [15]):

$$\Pr\left[G^{-1}(Y) \leqq x\right] = \Pr\left[Y \leqq G(x)\right] = G(x).$$

The above provides the basis for generating random numbers associated with an arbitrary distribution by transforming U(0,1) numbers.

In the case of the exponential distribution where $y = G(x) = 1 - e^{-x}$, it is possible to solve explicitly for x, that is, $x = -\log_e(1-y) \geqq 0$. In this case we say that G has been found <u>in closed form</u>, meaning that x has been expressed in terms of the logarithm function, one of the functions which, like the sine and cosine, we choose to call elementary functions.

For a general distribution function, it is not always possible to find the inverse function in closed form, that is, it is not always possible to express the inverse in terms of so-called elementary functions.

For the normal distribution, a closed form expression for the inverse function is not available. In such a case it may be possible to employ a graphical interpretation. Graphical methods, however, have a minimum of decimal place

accuracy and lead to table look-up procedures in high speed digital computers. Entering the ordinate of Figure 1 to determine abscissa values by use of a table, sometimes called the "Normal Curve of Error Table", also leads to table look-up methods in the computer.

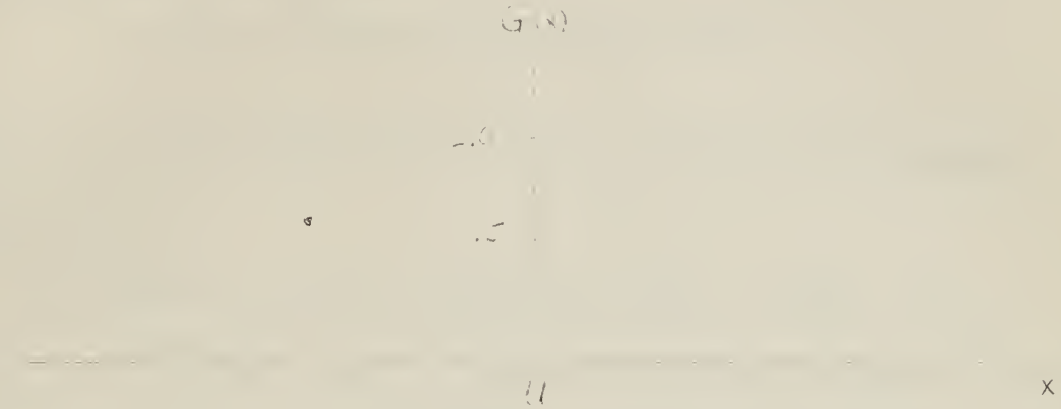

Figure 1

The disadvantage of the table look-up method of supplying random numbers for a war game is apparent when the memory space limitation of a computer is considered.

Even though no inverse function may exist in closed form, it is possible to construct generating schemes that are fast and have acceptable properties. Two such generating schemes for the normal distribution are given in the next two sections.

1.2 Normal Numbers by the Box-Muller Method (NORMGEN)

Box and Muller [1] proposed a method for generating a

pair of independent random N(0,1) numbers from a pair of random U(0,1) numbers. The transformation (see Appendix A) for generating the pairs of N(0,1) numbers is:

$$X_1 = (-2 \log_e U_1)^{\frac{1}{2}} \text{Cos } 2\pi U_2$$

$$X_2 = (-2 \log_e U_1)^{\frac{1}{2}} \text{Sin } 2\pi U_2$$

where U_1, U_2 are independent U(0,1) and X_1, X_2 are independent N(0,1).

In view of the four functions (sine, cosine, $\log_e$, square root) which must be computed in the Box-Muller method, this method cannot be considered fast, although a pair of numbers is produced so that computation time must be halved for comparison with other methods. The major advantage of the Box-Muller method is that it is an analytically exact transformation (of pairs of numbers) from the uniform distribution and will faithfully produce a normal distribution given a uniform distribution.

One disadvantage, other than computation time, is that the Box-Muller method will reflect whatever bias or contamination is present in the uniform generator.

When written as a subroutine and included in a service library, the Box-Muller normal generator may be employed in

two ways: The pair of N(0,1) numbers may be delivered simultaneously, X_1 in the A-register and X_2 in the Q-register, or the subroutine may save X_2 until the next call for a random number. In the first instance the user would be required to save X_2 for any case where only a single number were desired.

In general, standard library subroutines are very expensive in machine time, as they are required to be able to accept the general case.......they must test for sign and magnitude in many cases before beginning the computation. In special cases where the argument with which the subroutine is entered is $0 \leqq X \leqq 1$, or perhaps $0 \leqq X \leqq 2\pi$, the tests are not germane and precious machine time is wasted. In cases such as this the functional subroutines can be rewritten and shortened and included as part of the generator subroutine. However there is trade-off here, as this greatly increases the length of the generator and consequently the library program. The library function routines are then partially duplicated within the generator routine when the generator routine becomes part of the library. Of the four generators described in this thesis, these remarks pertain almost exclusively to the Box and Muller NORMGEN.

1.3 Normal Numbers from Sums of Uniforms (NORMSUM)

A discussion of the justification for approximation of a normal (0,1) random number by the sum of 12 uniform (0,1) random numbers appears in Appendix A. The approximation can be improved by increasing the number of uniform (0,1) numbers used, but machine time places a restriction on the desirability of using more numbers. The sums of uniform random variables converge more rapidly to a normal distribution than the sums of random variables from a non-symmetric distribution. The program for generating these numbers is short in both program steps and machine time. There are no library subroutines which must be called in.

This method is appealing to the intuition as there is the factor of "smoothing" or averaging whatever bias or contamination may be present in the uniform (0,1) generator.

1.4 Exponential Random Numbers (EXPOGEN)

Marsaglia's [2] method for generating exponential random numbers is more complicated in programming details than either the NORMGEN or NORMSUM, but lends itself very well to the FORTRAN language.

The expected value of the n of Marsaglia's method (see Appendix B) is approximately 1.58, so on the average only

1.58 U(0,1) numbers are required from which the minimum must be selected, and 1.58 discriminations (see Appendix B) are required to assign a value to n. Assigning a value to m also requires an average of 1.58 discriminations. Generation of an exponential number thus requires an average of approximately 4.75 U(0,1) numbers.

This method of producing exponential random numbers is sensitive to any bias in the characteristics of the uniform generator.

1.5 Chi-Square Random Numbers (CHIGEN)

Suppose that ten chi-square random numbers with five degrees of freedom are desired and that the NORMSUM generator is used to produce the normal numbers. It takes 12 uniform numbers to produce one normal number and five normal numbers to produce one chi-square number with five degrees of freedom. Thus we require 50 normal numbers and 600 uniform numbers to produce ten chi-square numbers with a relatively small number of degrees of freedom. One has no difficulty seeing that production of chi-square numbers is expensive in machine time. However, in war gaming the need for chi-square numbers is likely to be considerably less than the need for uniform or normal numbers.

The chi-square generator program is again relatively direct and no special difficulties are encountered. The frequency and goodness-of-fit tests do present a special problem, as changing the degrees of freedom in effect creates a new distribution and the class intervals and expected frequencies of occurrence within the class intervals change. They do not change in such a fashion that DO-LOOPS may be most economically employed. The program becomes quite lengthy in machine time if chi-square numbers with several different degrees of freedom are tested.

1.6 Computer Language

FORTRAN was the computer language employed for all programs written for this thesis. It is realized that FORTRAN may not be the best language for all types of war games, although successful war games have been written in this language. Speed and memory space limitations may dictate the use of assembly routines as opposed to FORTRAN-type compilers. However, within a FORTRAN program, strings of symbolic machine language may be inserted where the legal FORTRAN statements are not appropriate. In general, the particular character of the simulation will suggest the most convenient language.

For use in a war game, the four random number generators in this thesis must be translated into the language in which

the game is written. One word of caution is in order at this point. The method used to generate the U(0,1) numbers depends on the computer having a 48-bit word length, and the method must be revised for use on a machine with a different word length.

The Control Data Corporation 1604 stored program, general purpose digital computer with a storage capacity of 32,768 48-bit words was available at the Naval Postgraduate School for this project.

SECTION 2

TESTING OF RANDOM NUMBERS

2.1 General Background

The purpose of this section is to discuss the following question: When a generator has produced a sequence of random numbers and it is desired to associate them with a particular distribution function F, what should be the criteria for deciding on the merits of the generator?

For illustrative purposes the N(0,1) distribution will frequently be referred to in this section, although most of the remarks will pertain to the generation of any sequence of random numbers.

The criteria will to some extent depend on the employment of the generator. For example, statistical sampling experiments for estimating the parameters of mixed populations require very strict acceptance criteria for the random number generator. The employment considered in this paper will be primarily in the war game area.

If the F with which the sequence of random numbers is to be associated is N(0,1), it is obvious that a mean near zero and a variance near one must be demanded, but is this sufficient? The answer is emphatically no, as Figure 2 illustrates.

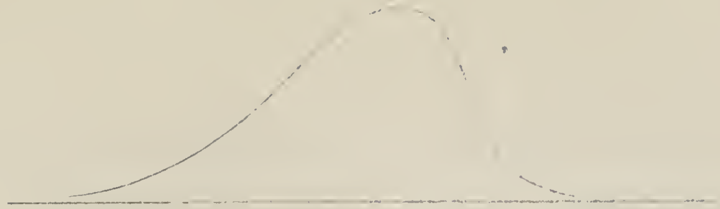

Figure 2

The density function graphed in Figure 2 could well have a mean of zero and a variance of one, but the skew to the left eliminates it from consideration.

A test of the sequence of numbers by computation of the first three central moments gives the criteria for the mean, variance and skew. Is this suffiecient? The answer is definitely no again. Two density functions may have the same values for the first three central moments and yet have quite different values for the fourth and higher central moments. Kurtosis is the state of "peakedness" or "flatness" of a density function. A rough measure of kurtosis is the standardized fourth central moment. A distribution having a relatively high peak such as the curve in Figure 3a is called leptokurtic, while the curve in Figure 3b which is flat-topped is called platykurtic. A distribution which is neither excessively peaked nor excessively flat is called "mesokurtic", for example the normal distribution, see [11].

Leptokurtic

Figure 3a

Platykurtic

Figure 3b

Now we have established a requirement for at least computing the first four central moments to provide data for deciding if a sequence of numbers can be regarded as a sample from a desired distribution, F.

At this point the experimenter would do well to apply a frequency test to the sequence of numbers by dividing the theoretical range of the numbers into class intervals and counting the numbers which fall into each class interval. Since the theoretical frequency of occurrence of numbers in each class interval is known, a goodness-of-fit test like the chi-square test may be applied to test for a significant difference between sample frequencies and observed frequencies in the class intervals.

In the case of the N(0,1) distribution, any sample considered to have been drawn at random should have an approximately equal number of positive numbers and negative numbers. A count of positive signs thus provides the data for another

test. The occurrence of positive signs for any distribution has the binomial distribution with $p = \frac{1}{2}$, if the median of the distribution is at zero. See Section 2.4.

The tests discussed up to this point were concerned with the distribution associated with the generated sequence of random numbers. The order in which the numbers in the sequence are generated is also of primary importance. For example, in the general case, a sample of say 10,000 numbers could be generated which would pass all six of the tests for a given distribution and yet have been produced as a monotonic sequence. Clearly the war gamer would not ordinarily be interested in this sample.

How then do we test for randomness, in contrast to testing for the characteristics of the distribution? Since we do not want the numbers in the sequence to have a predictable relationship to numbers following later in the sequence, a test for the serial correlation of the numbers (with various lags) is desirable.

The numbers may show a low serial correlation and yet come in excessively long runs of either positive or negative numbers. For the war gamer this could have disasterous effects, as his expected values could be based on a series of decision points where his random number generator is produc-

ing a run of all positive numbers, or all negative numbers. The over-all effect of this situation is that the game must be played a prohibitively large number of times to obtain results that have a meaningful average.

The above general testing information is summarized in Table 1. It is to be emphasized that these tests are necessary tests, but are not sufficient to guarantee the desired properties, even when administered collectively.

TABLE 1

TEST	NECESSARY	SUFFICIENT
MEAN	YES	NO
VARIANCE	YES	NO
SKEW	YES	NO
KURTOSIS	YES	NO
FREQUENCY	YES	NO
SIGN	YES	NO
SERIAL CORRELATION	YES	NO
RUN	YES	NO

The discussion of testing requires an explanation of the meaning given to n, the sample size. The multiplicative scheme used to generate the U(0,1) numbers used in the generators in this thesis has a period of 2^{45} numbers and produces the odd numbers between 1 and 2^{46}. These numbers are

transformed onto the interval (0,1). The U(0,1) numbers are transformed in one-to-one correspondence to numbers of another distribution, so a cycle in the generation of the U(0,1) numbers implies a cycle of the same period for the transformed numbers. A sample size of n = 100 means the <u>first</u> 100 numbers of the cycle and a sample size of n = 500 means the <u>first</u> 500 numbers of the cycle. In the second case the first 100 numbers are repeated as the first part of the sample of 500 numbers. In other words, all samples start with the first number of the cycle produced by the uniform generator.

Since some 8,000 billion numbers are produced before repeating, an experimenter could enter this cycle at any desired point and start generating numbers which suit his purposes. The tests in this thesis extend only to the first 10,000 numbers of the cycle because of the restrictions of available machine time. The generation of 10,000 normal numbers is quite rapid, but the battery of tests applied drastically increases the computer time required.

2.2 Test of Moments

The moment tests are constructed from the definitions of the first four central sample moments. All summations are taken on i, for i = 1,2,----,n.

$$m_1 = (\sum x_i)/n = \bar{x}$$

$$m_2 = \left[\sum (x_i - \bar{x})^2\right] /(n-1) = \left[\sum x_i^2 - (\sum x_i)^2/n\right] /(n-1)$$

$$m_3 = \left[\sum (x_i - \bar{x})^3\right] /(n-1)$$
$$= \left[\sum x^3 - 3(\sum x_i \sum x_i^2)/n + 2/n^2 (\sum x)^3\right] /(n-1)$$

$$m_4 = \left[\sum (x_i - \bar{x})^4\right] /(n-1)$$
$$= \left[\sum x_i^4 - 4(\sum x \sum x_i^3)/n + 6(\sum x)^2 \sum x_i^2/n^2 - 3(\sum x)^4/n^3\right] /(n-1)$$

m_1 through m_4 are computed from generated sample values and are consistent estimates of the theoretical moments.

All moments exist for the normal, exponential and chi-square distributions, and the theoretical values may be obtained from the moment generating functions. For the N(0,1) distribution the odd moments vanish and the second and fourth moments have a value of one and three, respectively. Comparisons of the first four sample moments with their theoretical values are made for all four random number generators in this paper.

Let

$$\beta_1 = m_3^2/m_2^3 \qquad\qquad \beta_2 = m_4/m_2^2$$

where m_2, m_3 and m_4 are the central moments defined above. β_1 and β_2 are called the coefficients of skew and kurtosis,

respectively [11]. The theoretical value of β_1 is zero and β_2 is three for the normal distribution. If β_2 is greater than three, the indication is that the density function is excessively peaked compared to the normal distribution, and if β_2 is less than three the indication is that it is excessively flat.

Programming of this test is slightly cumbersome, but straightforward, using the FORTRAN language. The computational factors $\sum x$, $\sum x^2$, $\sum x^3$, $\sum x^4$ are computed first.

For all tests of n numbers, the n numbers are generated consecutively and stored in a table. The tests then operate on the tabled numbers. When the generators are used as subroutines in war games they will not ordinarily be used to fill a table in this manner, but will deliver numbers for immediate use. For testing however, each number is used at least once for each test applied. If the numbers are not preserved in a table they must be regenerated for each successive test. Using a table to preserve the numbers limits the size of the sample that can be tested to something less than 20,000 numbers, but the largest sample tested in this paper is 10,000.

2.2 Chi-Square Analysis

Two chi-square goodness-of-fit tests are conducted on

the random $N(0,1)$ numbers generated by NORMGEN and NORMSUM. The difference lies in the choice of the class intervals. In the first case the abscissa is divided into 14 class intervals as shown in Figure 4. Twelve of the intervals are of equal length and the numbers greater than three are combined into a single class and the numbers less than minus three are combined into a single class.

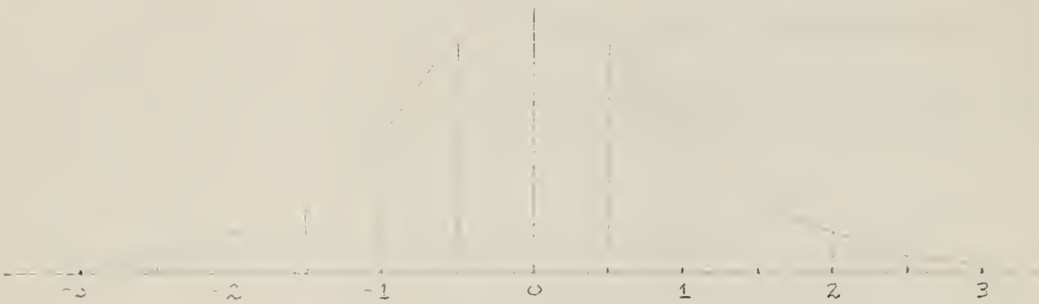

Figure 4

For the second test the class intervals are assigned equal probability of occurrence of sample values, rather than equal length subdivisions. The class intervals are chosen as indicated in Figure 5.

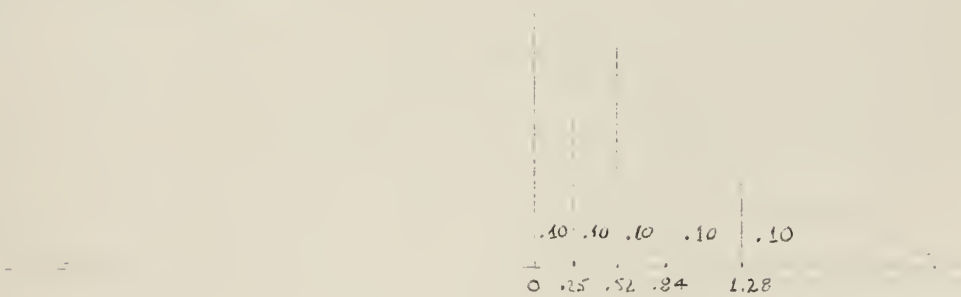

Figure 5

For the exponential and chi-square distribution goodness-of-fit tests, ten class intervals with equal probability were

tion.

As indicated by Mann and Wald [10], the sensitivity of the chi-square test for all distributions is greatly affected by the choice of class intervals. Two widely divergent assignments of the intervals were used in this paper to provide a better insight into deviations from the normal distribution. If a given sample passes both frequency tests or fails both tests the experimenter can be quite confident in his results, but if the sample fails one test and passes the other, the outcome is not so clear.

The chi-square statistic used was:

$$\text{Chi-square} = \sum_{i=1}^{k} (O_i - E_i)^2 / E_i \qquad \text{where}$$

O_i is the number of occurrences observed and e_i the number expected in the ith class interval.

2.4 Sign Test

This test is a count of the number of positive signs produced in a given sample and applies only to the normal distribution in this paper. The probability assigned to the occurrence of a positive sign is 0.5 and the distribution of occurrence is binomial. The smallest sample size considered is $n = 100$ where $npq = 25 >$ [illegible], so the normal approximation to the binomial is employed. See Bowker and Lieberman, pages [illegible] and

17[illegible] [1[illegible]]. The test statistic is:

$$z = \frac{x - np}{\sqrt{npq}},$$

and two-sided tests for significance at the 0.05 and 0.01 levels of significance are applied.

2.5 Serial Correlation Test

The serial correlation coefficient is a measure of the relationship of numbers in a sequence to numbers occurring later in the sequence. For example, given a sequence of n numbers $(U_1, U_2, ----, U_n)$, the relationship of the first number to the third, the second to the fourth, the third to the fifth, and so on is called serial correlation, lag two, and is defined by:

$$r_2 = \frac{\mathrm{Cov}\,(U_j, U_{j+2})}{\sqrt{(\mathrm{Var}U_j)(\mathrm{Var}U_{j+2})}}$$

In general, summing on j, j = 1,2,----,n-k:

$$r_k = \frac{\frac{1}{n-k}\sum U_j U_{j+k} - \frac{1}{(n-k)^2}\sum U_j \sum U_{j+k}}{\left[\frac{1}{n-k}\sum U_j^2 - \frac{1}{(n-k)^2}\left(\sum U_j\right)^2\right]^{\frac{1}{2}}\left[\frac{1}{n-k}\sum U_{j+k}^2 - \frac{1}{(n-k)^2}\left(\sum U_{j+k}\right)^2\right]^{\frac{1}{2}}}$$

Although this computation formula is rather involved,

it lends itself nicely to FORTRAN programming. It must be remembered, however, that $r_0 = 1$, and r_{n-1} and r_n are undefined.

The value of r_k ranges from plus one to minus one, with zero the ideal value. It is desired to keep the absolute value of r_k low without long runs of positive and negative signs.

2.6 Run Test

The test described in this section will only be applied to numbers generated by [illegible] and [illegible], although a modification of this test could be easily applied to [illegible] and [illegible]. See [16].

In any sequence of random N(0,1) numbers a certain number of runs of positive and negative numbers will occur by chance. If the numbers generated were alternately positive and negative with no runs greater than one in length above or below zero, one could hardly say that these numbers were random, as the fit would be too good. The greatest number of runs should be of length one, the next greatest of length two and so forth. In a given sample the runs may be counted and compared with the theoretical values for determining significant differences.

As described by Ernest J. Lytle, Jr. of the University

of Florida [4], the expected number of positive and negative runs of length i from n numbers is approximately:

$$\frac{n}{2^{i+1}} \qquad i = 1,2,-----,n$$

Since it is desired to keep the expected number of runs in any class interval greater than some number, say 10, for a goodness-of-fit test, the expected numbers of runs greater than a certain length may be grouped into a single class. The expected number of positive and negative runs of length greater than or equal to some number k is approximately:

$$\frac{n}{2^k}$$

Let K_α be defined by the following equation:

$$\Pr\left[\text{at least one run of length} \geqq K_\alpha\right] = \alpha$$

It is shown in [16] that $K_{.05}$ is approximately:

$$2.3(\log_{10} n + 1)$$

SECTION 3

TEST RESULTS

The results of the tests performed on sequences of numbers generated by NORMGEN, NORMSUM, EXPOGEN and CHIGEN are given in Tables 2-9. It should be noted that Pos Corr. and Neg Corr. in Tables 2-5 refer to the largest positive and largest negative values of serial correlation obtained in tests of lag one through 20.

For the normal generators in Tables 2 and 3, CHISQR1 refers to a frequency test with equal class intervals and 13 degrees of freedom, and CHISQR2 refers to a frequency test with equal expected frequencies in class intervals and nine degrees of freedom. The theoretical values of chi-square are the values above which the test results are significant at the .05 level. NORMGEN and NORMSUM both passed the goodness-of-fit tests for runs in samples of size 1000 and greater at the .05 level of significance as indicated in Table 9, and there were no runs longer than $K_{.05}$, where $K_{.05} = 3.3(\log n + 1)$ as defined in Section 2.

As EXPOGEN fails to pass the frequency tests for sample sizes of 5000 and 10,000 at either the .05 or .01 significance levels, the frequency count for the ten class intervals for samples of size 1000, 5000 and 10,000 are compared in Table 6.

Although the percentages of numbers in class intervals are similar, the fit becomes poorer as the sample size increases due to the squared difference in the numerator of the chi-square statistic. For example, 120 x 120 = 14,400 is 100 times as great as 144, while 10,000 (the denominator) is only ten times greater than 1000. Also it is noted that a sample of size 100 fares very badly in most tests performed on sequences of numbers produced by EXPOGEN. The magnitude of exponential numbers is very sensitive to the magnitude of the uniform numbers from which they are transformed. This is true of Marsaglia's method [2] and a more well-known method, i.e., taking the logarithm of U(0,1) numbers.

The last column of Table 5 (CHIGEN) gives the values of the chi-square goodness-of-fit statistic using 10 classes. For nine degrees of freedom the .95th and .99th percentile points of the chi-square distribution are 16.9 and 21.7 respectively. Thus, significant results at the 5% level were obtained for n = 200 and ν = 3,4,5 and 15. In order to investigate this further a chi-square analysis was done for samples of size 500 and 1000 (Table 7). It should be noted that fit does not necessarily improve with sample size. For all degrees of freedom and sample sizes considered, CHIGEN passes the test of fit at the .01 level of significance.

To further explore the properties of the CHIGEN numbers, the first four moments were computed but not tabled for samples of size 100, 500, 1000 and 5000 with five degrees of freedom. The theoretical values of the first four moments are 5,10, 40 and 540, respectively, as computed from the moment generating function. The values of the first and second moments were reasonably close to the theoretical values, but the values of the third moments were quite low for all sample sizes. The value of the fourth moment was low for a sample size of 500, indicating a possible "flat" curve for the density.

Some of the raw data test results have been included in Tables 2-[illegible], as they are likely to be of interest to experimenters. Table [illegible] is a summary of results of tests for n = 1000 for CHIGEN and n = 10,000 for all other generators. The terms GOOD, FAIR and POOR in Table [illegible] are used in an intuitive and relative sense rather than as any absolute standard. GOOD generally implies that the theoretical and test values are not significantly different at the .05 level and no undesirable characteristics are apparent. POOR implies significant differences at the .01 level or sufficient deviations from theoretical values to make the generator suspect for some applications. FAIR implies a condition somewhere in between the two extremes.

Table 2 - NORMGEN

	n=100	n=500	n=1000	n=5000	n=10000	Theor.
Mean	-.005	.034	-.015	-.009	.015	.00
Variance	1.719	1.074	1.030	1.026	1.021	1.00
3rd Moment	-2.77	-.741	-.270	-.054	-.036	.00
4th Moment	12.5	[illegible].006	4.423	3.223	3.131	3.00
β_1 (Skew)	1.51	.4079	.0667	.0127	.0012	.00
β_2 (Kurtosis)	6.606	5.204	4.167	3.166	3.042	3.00
Pos. No's	51	257	497	2506	5071	50%
CHISQ1	20.15	12.43	12.39	13.09	13.09	22.4
CHISQ2	10.00	15.24	20.94	10.69	11.78	16.9
Max Pos.	2.46	2.74	2.74	3.63	3.68	
Max Neg.	-5.77	-5.77	-5.77	-5.77	-5.77	
Pos Corr.	.206	.123	.059	.032	.026	.00
Neg Corr.	-.262	-.145	-.070	-.035	-.018	.00

Table 3 - NORMGUN

	n=100	n=500	n=1000	n=5000	n=10000	Theor.
Mean	.035	-.077	-.055	-.033	-.012	.00
Variance	1.022	1.117	1.127	1.023	1.003	1.00
3rd Moment	.049	-.022	.053	.019	-.002	.00
4th Moment	2.60	3.32	3.20	3.07	3.07	3.00
β_1 (Skew)	.00190	.00035	.00234	.00031	-.00003	.00
β_2 (Kurtosis)	2.076	2.662	2.590	2.973	2.979	3.00
Pos. No's	48	234	466	2425	4945	50%
CHISQR1	4.33	16.35	36.05	30.26	35.32	22.4
CHISQR2	3.20	14.48	26.90	23.13	20.22	16.9
Max Pos.	2.91	2.91	2.91	3.35	3.40	
Max Neg.	-2.92	-2.97	-2.97	-4.31	-4.31	
Pos Corr.	.265	.115	.073	.035	.016	.00
Neg Corr.	-.199	-.096	-.070	-.023	-.013	.00

Table 4 - EXPOGEN

	n=100	n=500	n=1000	n=5000	n=10000	Theor.
Mean	.958	1.020	1.021	.988	1.002	1.000
Var.	.636	1.040	.971	.992	1.036	1.000
3rd Moment	.608	1.992	1.536	1.925	2.169	2.000
4th Moment	4.620	3.446	6.156	8.191	10.061	9.000
Chi-sqr	13.0	6.72	9.70	22.58	33.70	16.90
Largest	3.67	7.18	7.18	8.32	10.57	
Pos Corr	.215	.107	.043	.035	.017	
Neg Corr	-.267	-.072	-.051	-.028	-.027	

Table 5 - CHIGEN (n=200)

DF(ν)	Mean	Var	PosCorr	NegCorr	Largest	Chi-sqr
1	1.187	2.352	.072	-.118	7.94	8.42
2	2.153	3.977	.111	-.115	9.85	7.39
3	3.382	6.001	.132	-.103	11.12	18.34
4	4.564	7.671	.149	-.136	11.59	19.07
5	5.649	10.321	.095	-.129	15.71	19.42
10	10.733	17.872	.122	-.143	23.54	12.33
15	15.980	24.293	.105	-.168	28.21	19.21
20	21.120	40.395	.093	-.094	41.32	14.65
Theor.	ν	2ν				16.90

Table 6 - FREQUENCY COUNT FOR TIPOGEN NUMBERS

Interval	n=1000	n=5000	n=10000	Theor.
1	98	522	1037	10%
2	96	509	977	10%
3	97	487	995	10%
4	114	544	1046	10%
5	83	472	973	10%
6	97	487	957	10%
7	111	556	1120	10%
8	87	434	906	10%
9	106	487	951	10%
10	111	502	1038	10%

Table 7 - CHI-SQUARE ANALYSIS FOR CHIGEN

	n=200	n=500	n=1000
1	9.42	13.81	10.57
2	7.39	20.33	10.25
3	13.34	17.06	17.39
4	19.07	15.04	13.46
5	10.42	17.01	14.99
10	12.83	8.94	14.57
15	13.21	17.87	18.16
20	14.65	10.86	18.40

Table 8

CHI-SQUARE VALUES FOR RUN TESTS

	Computed Values		Theoretical Values		
n	NORSIGN	NORMSUM	DF	.05	.01
100	15.34	4.02	2	5.99	9.21
500	6.39	10.50	4	9.49	13.3
1000	3.22	6.30	4	9.49	13.3
5000	7.26	[illegible]	8	15.5	20.1
10000	[illegible]	6.40	9	16.9	21.7

Table 9 - SUMMARY

TEST	NORSIGN	NORMSUM	CHIGEN	EXPOGEN
MEAN	GOOD	GOOD	GOOD	GOOD
VARIANCE	GOOD	GOOD	FAIR	GOOD
SKEW	GOOD	GOOD	POOR	GOOD
KURTOSIS	GOOD	POOR	FAIR	FAIR
FREQUENCY	GOOD	FAIR	FAIR	FAIR
SIGN	GOOD	GOOD	NA	NA
CORRELATION	GOOD	GOOD	GOOD	GOOD
RUNS	GOOD	GOOD	GOOD	GOOD

SECTION 4

CONCLUSIONS

The tests have demonstrated that random numbers may be generated that behave when tested like numbers from a normal, exponential or chi-square distribution. It is apparent, however, that the behavior of these numbers depends heavily on the multiplicative scheme used to generate the U(0,1) random numbers from which they were transformed. Some of the generating schemes discussed in this thesis are theoretically exact so that the tests described in effect provide additional tests of the performance of the random U(0,1) generator that is an integral part of NORMGEN, NORMSUM, EXPOGEN and CHIGEN. See [6] for generation and testing of U(0,1) random numbers.

In general, the test results of Section 3 indicate that NORMGEN and EXPOGEN must be used with caution for very small samples and NORMSUM must be used with caution for large samples. The chi-square generator was obtained by two transformations and should be carefully examined for the sample sizes and degrees of freedom required, as the results are not consistent throughout. Further testing is required for the latter and will require rather extensive machine time if large samples and several numbers of degrees of freedom

are considered.

A model (game, experiment) which uses a great many random numbers should have the capability of testing samples of these numbers at desired intervals to ensure that the number generator has not fallen into some sort of trap such as finding a zero in a multiplicative scheme. This, of course, is merely an extension of the general rule of not trusting the machine output answers before submitting the model to a test program with known answers. In any extensive program the arithmetic schemes may be giving answers to questions that were really asked, and not to questions the programmer thought he asked.

As an incidental observation, it is believed that the logical sequence of learning program techniques is to start with the machine language and the fundamental machine instructions, progress to an assembler and finally to a compiler. Even though a compiler language such as FORTRAN is advertised as "no computer experience necessary", the programmer who does not understand the basic machine instructions will find de-bugging his program a nearly impossible task. This is apparently not a matter of general agreement.

Since random numbers have been generated by several means other than arithmetic schemes, including generation by

mechanical and electrical devices [14], it would seem reasonable to make a comparison. Two or three of the "best" published tables of random numbers could be typed onto cards and inserted into machine memory for testing by the same battery of tests that NORGEN, NORMSUM, CHIGEN and EXPOGEN were subjected to. It is realized that typing 10,000 IBM cards is not a small task. Prepared decks of cards may be available in some cases.

The iterative, multiplicative algorithm used to generate the U(0,1) sequence delivers a long cycle of numbers without repeating individual numbers. Blocks of these numbers within the cycle are quite likely to have undesireable characteristics. The cycle can be entered at any point by inserting the desired "priming number" in the algorithm. If small blocks of numbers were tested, those blocks showing poor characteristics could be dropped from the cycle by testing for the critical priming numbers. There is a definite danger involved in this procedure. The dropping of short blocks of numbers showing poor characteristics may improve small samples and degrade the characteristics of large samples. This matter has apparently not been discussed in the literature.

BIBLIOGRAPHY

1. Box, G. E. P. and Muller, M. E. — A NOTE ON THE GENERATION OF RANDOM NORMAL DEVIATES. Anals of Mathematical Statistics, Volume 29, Number 2, June, 1958

   Presentation of the NORMGEN method of this thesis and suggestions for producing chi-square, F and t distributions.

2. Marsaglia, G. — GENERATING EXPONENTIAL RANDOM VARIABLES. Anals of Mathematical Statistics, Volume 32, Number 3, September, 1961

   Offering a device for producing exponential random variables by performing discriminations on the relative magnitudes of uniform (0,1) random variables.

3. Marsaglia, G. — EXPRESSING A RANDOM VARIABLE IN TERMS OF UNIFORM RANDOM VARIABLES. Anals of Mathematical Statistics, Volume 32, Number 3, September, 1961

   Suggestion for expressing a distribution function as a mixture of suitably chosen distribution functions .... choosing a distribution function which is close to the original and using it most of the time, applying the correction only infrequently.

4. Lytel, E. J., Jr. — A DESCRIPTION OF THE GENERATION AND TESTING OF A SET OF RANDOM NORMAL DEVIATES. Symposium on Monte Carlo Methods, University of Florida, Wiley, 1956

   Generation of normal deviates employing numerical methods and details of tests for normality and randomness.

5. Kendall, M. G. RANDOMNESS AND RANDOM SAMPLING NUMBERS
Journal of Royal Statistical Society,
101 (1938)

A detailed discussion of randomness and tests for randomness.

6. Harling, John SIMULATION TECHNIQUES IN OPERATIONS RESEARCH - A REVIEW
ORSA Journal
May-June, 1958

As the title indicates.

7. Bauer, W. F. THE MONTE CARLO METHOD
J. Soc. Indust. Appl. Math.
Volume 6, Number 4, December, 1958

A general discussion covering broad areas. A very long bibliography is appended.

8. Mood, A. M. INTRODUCTION TO THE THEORY OF STATISTICS
McGraw-Hill
First Edition 1950

A standard textbook.

9. Barron, J. M. RANDOM NUMBER GENERATION ON DIGITAL COMPUTERS
U.S. Naval Postgraduate School
Thesis, 1962

On selecting parameters in a multiplicative congruential method for the CDC 1604 computer.

10. Mann, H. B. and Wald, A.
ON THE CHOICE OF THE NUMBER OF CLASS INTERVALS IN THE APPLICATION OF THE CHI SQUARE TEST
Anals of Mathematical Statistics
Volume 13, 1942

A detailed discussion of the criteria for selecting the number and length of class intervals.

11. Kendall, M. G.
THE ADVANCED THEORY OF STATISTICS
Volume I, Chapters 1 through 16
Charles Griffin & Company, Ltd., London
Third Edition, 1947

A comprehensive textbook, intended to be a book on statistics, not on statistical arithmetic.

12. Kendall, M. G.
THE ADVANCED THEORY OF STATISTICS
Volume II, Chapters 17 through 30
Hafner Publishing Co., New York
Third Edition, Third Impression, 1959

A continuation and expansion of Volume I with a bibliography of more than 1500 titles. The two volumes comprise a treatise on statistics.

13. Bowker, A. H. and Lieberman, G. J.
ENGINEERING STATISTICS
Prentice-Hall, Inc.
Fourth Printing, May, 1961

Intended as a text for a first course in statistics for students in engineering and the physical sciences.

14. RAND Corporation
A MILLION RANDOM DIGITS WITH 100,000 NORMAL DEVIATES
The Free Press
Glencoe, Illinois, 1955

As the title indicates.

15. Parzen, E. MODERN PROBABILITY THEORY AND ITS APPLICATIONS
John Wiley & Sons, Inc., 1960

An introductory textbook on discrete and continuous probability with a calculus prerequisite.

16. Wald, A. STATISTICAL THEORY WITH ENGINEERING APPLICATIONS
John Wiley & Sons, Inc.

A statistical technique reference book for students with a calculus background. Oriented toward engineering with many applications.

APPENDIX A

The purpose of this appendix is to discuss in more detail the two methods (NORMGEN and NORMSUM) of generating random N(0,1) numbers introduced in Section 2.

It is assumed, as has been done throughout this thesis, that a method to generate uniform random numbers is available. A transformation of uniform random numbers to normal random numbers will now be considered. Of the several existing methods to accomplish this, the method suggested by Box and Muller [1] was selected for accuracy and integrity in the tails of the distribution, as war gamers and others can be quite interested in "rare events".

<u>NORMGEN</u>: Let U_1, U_2 be independent random variables from the uniform density function on the interval (0,1). Consider the random variables:

$$(1) \quad \begin{aligned} X_1 &= (-2 \log_e U_1)^{\frac{1}{2}} \operatorname{Cos} 2\pi U_2 \\ X_2 &= (-2 \log_e U_1)^{\frac{1}{2}} \operatorname{Sin} 2\pi U_2 \end{aligned}$$

Then (X_1, X_2) are a pair of independent random variables from the same normal distribution. Notice that the radical is always positive since $0 \leq U_1 \leq 1$.

<u>Justification</u>: Solving (1) for the inverse relation-

since we obtain without difficulty:

$$U_1 = \exp\left[-(X_1^2 + X_2^2)/2\right]$$

$$U_2 = -(1/2\pi)\arctan(X_2/X_1) .$$

The Jacobian of (U_1, U_2) with respect to (X_1, X_2) is found to be $(1/2\pi)\exp\left[-(X_1^2 + X_2^2)/2\right]$ and the joint density function of U_1, U_2 is 1, so that the joint density function of X_1, X_2 is:

$$\begin{aligned} f(X_1, X_2) &= (1/2\pi)\exp\left[-(X_1^2 + X_2^2)/2\right] \\ &= (1/2\pi)^{\frac{1}{2}}\exp\left[-X_1^2/2\right] \cdot (1/2\pi)^{\frac{1}{2}}\exp\left[-X_2^2/2\right] \\ &= f(X_1) \cdot f(X_2) \end{aligned}$$

establishing that X_1 and X_2 are normal (0,1) and independent.

Expression (1) is analytically exact and the characteristics of the numbers produced are distorted from the ideal only by the pseudo random uniform numbers used as input and by the machine calculations, including the square root, log, sine and cosine routines. Expected accuracy is 13 decimal digits in the CDC 1604 FORTRAN routines. It is hoped and expected that computational round-off errors will be random.

<u>NORMSUM</u>: As some users may not be interested in rare events but in numbers around the mean we next select for

testing a method based on the Central Limit Theorem. The distribution of sums of independent random variables from symmetric distributions approaches the normal distribution quite rapidly. It is noted that 50 numbers, each composed of the sum of five random digits, gives a strikingly normal appearance when plotted in a histogram.

Since the variance of the U(0,1) distribution is 1/12 and variances of uniform random variables are additive under convolution it is convenient to select 12 as the number of uniform random variables whose sum will approximate a normal random variable. Means of uniform variables are also additive so that it remains to subtract the constant six from the sums of 12 independent U(0,1) random variables to approximate a normal (0,1) distribution.

This process truncates the distribution so that $-6 \leq x \leq 6$. How much is lost by this truncation can be estimated by the fact that normal tables with 4-place significance have positive probability within the limits $-4.26 \leq x \leq 4.26$.

There is a case for the truncated normal distribution in that a model is useful when it can be realistically applied to a real world situation. Consider the height of men of a given race as a random variable, normally distributed. For use the distribution is truncated in that the smallest

height is certainly greater than one foot and the greatest height less than ten feet. Definite bounds can be set on many errors of measurement which are considered normally distributed. Applications of the theoretical normal distribution are so frequently truncated in the real world that it seems inconsistent to reject a manually generated sample from a normal population if the sample exhibits the properties of the theoretical distribution between the limits of usable values.

APPENDIX B

EXPONENTIAL RANDOM NUMBERS

The exponential distribution figures prominently in particle or radiation studies, reliability, life testing and like areas which can be of interest to persons using high speed computers and monte carlo techniques. A power series expansion to compute the logarithm of a uniform random variable may be used to generate exponential random variables, but a more rapid method is desired.

G. Marsaglia of the Boeing Scientific Research Laboratories offers a simpler device for producing exponential random variables by performing discriminations on the relative magnitudes of uniform (0,1) random variables. See [2].

The idea is to choose the minimum of a random number of uniform random variables, then add a random integer---say, let n and m be random integers according to the following schedule:

Value of n	Prob.	Total	Value of m	Prob.	Total
1	.58	.58	0	.63	.63
2	.29	.87	1	.23	.86
3	.10	.97	2	.09	.95
4	.02	.99	3	.03	.98
:	:	:	:	:	:

Then if $U_1, U_2, U_3, \ldots\ldots\ldots$ is a sequence of independent uniform random variables on (0,1), the random variable

$$x = m + \min(U_1, U_2, \ldots\ldots U_n)$$

has the exponential distribution.

<u>Justification</u>: Let n be a random variable taking values 1,2,3,.... with probabilities $p_1, p_2, p_3, \ldots\ldots\ldots$. If

$$y = \min(U_1, U_2, \ldots\ldots U_n)$$

then the distribution of y is, for $0 \leqq \theta \leqq 1$,

$$P\left[y \leqq \theta\right] = 1 - p_1(1-\theta) - p_2(1-\theta)^2 - \ldots\ldots$$

In particular, if $c = 1/(e-1) = .5819767 \ldots$ and

$$p_1 = c, \quad p_2 = c/2!, \quad p_3 = c/3!, \ldots ,$$

then

$$P\left[y \leqq \theta\right] = ce(1-e^{-\theta}), \quad 0 \leqq \theta \leqq 1.$$

<u>Theorem</u>: If $c = 1/(e-1)$ and if the random variable n takes values 1,2,3, ... with probabilities c, c/2!, c/3!, ... and if, independently, the random variable m takes values 0,1,2, ... with probabilities $1/(ce)$, $1/(ce^2)$, $1/(ce^3)$, ..., then the random variable

$$x = m + \min(U_1, U_2, \ldots, U_n)$$

has the exponential distribution,

$$P\left[x \leqq t\right] = 1 - e^{-t}, \quad 0 \leqq t.$$

The proof is a matter of verification---if $x = k + \theta$

where k is a non-negative integer and $0 \leqq \theta \leqq 1$, then

$$P\left[X \leqq k + \theta\right] = P\left[m \leqq k-1\right] + P\left[m = k, y \leqq \theta\right]$$

$$= 1-e^{-k} + \left[ce/(ce^{k+1})\right] (1-e^{-\theta})$$

$$= 1-e^{-(k+\theta)}.$$

APPENDIX C

CHI SQUARE RANDOM NUMBERS

Let $x_1, x_2, \ldots, x_\nu$ be ν independent random variables normally distributed, each with zero mean and unit variance. Let χ^2 equal the sum of the squares of these ν random variables, i.e., $\chi^2 = x_1^2 + x_2^2 + \ldots + x_\nu^2$. χ^2 is also a random variable because it is a function of random variables. The density function of the chi-square random variable is written as

$$f(\chi^2) = \frac{1}{2^{\frac{\nu}{2}}\Gamma(\frac{\nu}{2})}(\chi^2)^{\left(\frac{\nu-2}{2}\right)} e^{-\frac{\chi^2}{2}} \qquad \chi^2 \geqq 0$$

$$= 0 \qquad \text{otherwise}$$

All x's must have zero mean and unit variance. If they are independent normal (μ, σ) numbers, the quantities

$$Z_i = \frac{x_i - \mu_i}{\sigma_i} \qquad i = 1, 2, \cdots, \nu$$

do have a normal distribution, each with zero mean and unit variance. Hence, summing on i, for i = 1,2,----,ν

$$\chi^2 = \sum Z_i^2 = \sum_i (x_i - \mu_i)^2/\sigma_i^2$$

has a chi-square distribution with ν degrees of freedom.

An example of such a random variable is as follows: A missile is sent toward a target. If the range and deflection errors of the missile are independent normally distributed random variables each with zero mean and equal variances, the squared distance between the explosion and the target, divided by the variance, has a chi-square distribution with two degrees of freedom. This can be easily extended to three dimensions.

A subroutine available to a war gamer to supply independent χ^2 numbers with arbitrary degrees of freedom thus can have immediate application.

APPENDIX D

PROGRAMS

Following is the Box-Muller method of generating pairs of N(0,1) numbers from pairs of U(0,1) numbers, programmed in the Control Data Corporation 1604 FORTRAN language.

The program as written merely computes 100 numbers and stores them in an array called "TAB". N is the sample size and M must always be one-half the value of N. The dimension of TAB must be at least the value of N. The numbers that the program stores in TAB are in floating point format.

```
PROGRAM NORMGEN
DIMENSION IU(2),TAB(100)
MASK=4000000000000000B
PI2=2.*3.14159265
UNIT=2.**23+3.
IUNIT=UNIT
IU(2)=1
N=100
DO 30 M=1,50
ENI2(1),ENI1(2),LDA1(IU),MUI(IUNIT),SCL(MASK),STA2(IU),
MUI(IUNIT),
SCL(MASK),
STA1(IU).
U1=IU(1)
U1=U1/2.**47
U2=IU(2)
U2=U2/2.**47
XLOG=LOGF(1./U1)
ARG=PI2*U2
SIN2=SINF(ARG)
COS2=COSF(ARG)
ROOT=SQRTF(2.*XLOG)
X1=ROOT*COS2
X2=ROOT*SIN2
K=2*M
```

```
      L=M-1
      TAB(M)=X2
      TAB(L)=X1
   30 CONTINUE
```

The CDC 1604 program for generating N(0,1) numbers from the sums of 12 uniform numbers also stores N numbers in "TAB".

```
      PROGRAM NORMSUM
      DIMENSION TAB(100)
      N=100
      MASK=4000000000000000B
      UNIT=2.**23+3.
      IUNIT=UNIT
      IU=1
      DO 702 I=1,N
      STOR=0
      DO 701 M=1,12
      LDA(IU),MUI(IUNIT),SCL(MASK),STA(IU).
      XIU1=IU
      XIU1=XIU1/2.**47
  701 STOR=STOR+XIU1
      STOR=STOR-6.
  702 TAB(I)=STOR
```

The chi-square generator has two parameters, the size of sample N and degrees of freedom NU.

```
      PROGRAM CHIGEN
      DIMENSION TAB(100)
      MASK=4000000000000000B
      UNIT=2.**23+3.
      NU=5
      N=100
      IUNIT=UNIT
      IU=1
      DO 803 I=1,N
      STOR1=0
      DO 802 IT=1,NU
      STOR=0
      DO 801 M=1,12
      LDA(IU),MUI(IUNIT),SCL(MASK),STA(IU).
      XIU1=IU
```

```
      IUI=[illegible]
  901 STOR=[illegible]
      STOR=[illegible]
      STOR=[illegible]
      STOR1=STOR1+STOR
  902 CONTINUE
      TAB(I)=STOR1
  903 CONTINUE
```

The exponential generator has the single parameter N.

```
      PROGRAM EXPGEN
      DIMENSION TAB(100),TEST(30)
      MASK=2000000000000000B
      MASK2=4000000000000000B
      E=2.718281828459
      UNIT=2.**23+3.
      N=100
      IUNIT=UNIT
      CX=1./(E-1.)
      IU=1
      DO 143 L=1,N
      S=CX
      LDA(IU),
      MUI(IUNIT),
      SCL(MASK2)
      STA(IU)
      ARS(11)
      ADD(MASK)
      FAD(MASK)
      STA(X)
      S=0.
      IXX=1
  134 XI=IXX
      C=C/XI
      S=S+C
      IF(S-X)133,135,135
  133 IXX=IXX+1
      GO TO 134
  135 LDA(IU)
      MUI(IUNIT)
      SCL(MASK2)
      STA(IU)
      ARS(11)
      ADD(MASK)
```

```
      FAD(MASK)
      STA(X)
      D=1./CX
      H=0.
      XM=0.
  137 D=D/E
      H=H+D
      IF(H-X)136,138,138
  136 XM=XM+1.
      GO TO 137
  138 LDA(MIN)
      SAU(139)
      ENI2(1)
 1381 LDA(IU)
      MUI(IUNIT)
      SCL(MASK2)
      STA(IU)
      ARS(11)
      ADD(MASK)
      FAD(MASK)
      STA2(TEMP)
  139 ISK2(X)
      SLJ(1381)
      DO 141 M=2,KKK
      IF(TEMP(1)-TEMP(M))141,141,142
  142 TEMP(1)=TEMP(M)
  141 CONTINUE
  143 TAB(L)=TEMP(1)+XM
```

The CDC 1604 computer test programs written for use in this thesis are available in FORTRAN language in the Computer Center of the U.S. Naval Postgraduate School, Monterey, California.